KIM-1/6502 USER NOTES ISSUE #12

## THIS IS YOUR LAST ISSUE !!!

## RENEWAL TIME IS NOW!!!

Since starting this newsletter several years ago, I've had the chance to communicate with many of you. One thing sort of held true through most of the conversation. Most of you wanted more information more often.

But, since "User Notes" was always a part time activity, it had to play second fiddle to my full time career. As a result, the "Notes" was late a good deal of the time. The situation was unfortunate, but there didn't seem to be a solution.

The past several months I have tried to devise means for expanding "User Notes" so as to provide a better service to you.

I have come to one conclusion. In order to do justice to the general readership, I have decided to make "User Notes" my full time activity. Now I'll be able to spend ALL my time doing a job which needs to be done. I have decided to continue being a bi-monthly publication - at least for a while - but expanding each issue to 24 pages - (double the size of this issue). We're going to continue with First Class mailing (it's faster) and are going to mail each issue in an envelope to eliminate lost pages and frustrated readers.

You'll also notice some big changes between the covers - WE'RE GOING TO SUPPORT VIM & AIM SYSTEMS, (as well as others).

Users of these other "soon-to-be-popular" 6502 based machines will need a place where they can exchange information and our "new" publication can gear up to the task.

With all these changes, it's only fitting that we have a new name to signify our new personality - from now on we'll be called "USER NOTES: 6502".

Our new address is:

USER NOTES: 6502
P.O. Box 33093
N. Royalton, Oh 44133

The new subscription rates will be:
$13.00 / 6 double issues - mailed 1st Class to USA & Canada
$19.00 / 6 double issues - Air Mailed overseas

US FUNDS ONLY
NO PURCHASE ORDERS WITHOUT PAYMENT PLEASE

If you have already resubscribed for Volume 3 at the old price and don't wish to continue your subscription, let us know - we'll cheerfully refund your money. If, on the other hand, you feel as we do that the best is yet to come, kindly remit enough funds to make up the difference.

## PC '78

If you got to PC '78 in Philadelphia your probably still thinking about some fo the neat things that were there. There certainly were a number of things to keep you entertained.

Hal Chamberlain, of MTU, was there with a pre-production copy of their new 16K dynamic RAM board. ($375).

They certainly seem to know the secrets of using dynamic ram up there at MTU.

Many of you have probably heard Hal's digital-to-analog converter board playing the Star Spangled Banner and sounding like a Hammond Organ.

They also showed their prototyping card and a card file which positioned the KIM horizontally above slots for 4 additional cards.

Chamberlain mentioned that since his dynamic memory and video board draws such a small amount of power, he can power two 16K RAM cards and one visable memory board from his $30 power supply.

Hudson Digital Equipment had two disc-based KIM systems up and running to show off their 6502 software and KIM expansion product line.

The most excitement at the HDE booth was the introduction of their KIM MINI-FLOPPY SYSTEM.

For $695.00, according to HDE, you'll get a Shugart drive, the 4.5"x6" controller board, all necessary cables and the software to drive the thing from your KIM system.

The software is a slightly scaled down version of FOD5 (file oriented disc system) which is included with their full size disc system. (I've been using this software for about six months and am quite impressed with its capability). A dual drive version of mini-floppy system drive will also be available but no price was mentioned.

They were also excited about their NEC/DIABLO interface hardware and software driver with right print justification.

(It would sure be fine to compose this newsletter on a video terminal and then print it on the NEC printer).

HDE also showed a very compact 4.5"x6" card rack, and a prototyping card for their system.

Another KIM-4 bus supporter, RNB Enterprises, (2967 West Fairmount Ave., Phoenix, Az 85017, 602-265-7564),was present with the VIM-1, from Synertek, and a KIM-VIM-AIM compatible motherboard together with RAM, EPROM & EPROM burner cards.

Their motherboard includes an aluminum card cage, can handle up to 8 KIM-4 compatible expansion cards, and sells for $139.95.

Also on display at the RNB booth was a 16K static RAM board ($379) using 2114's, a 2708 EPROM burner board ($269) and an EPROM carrier board ($129) for 2708, 2758, 2716 and 2516 EPROM's.

I'm really glad to see RNB & HDE supporting the KIM-4 bus. It makes alot of sense to support a bus which is so easy to design around.

Overall, PC '78 was great fun. Hope you got to see it.

**********

HERE'S THE REST OF THAT EXCELLENT GRAPHICS SERIES STARTED SEVERAL ISSUES AGO BY [illegible] FLACCO.

SCOPE LUNAR LANDER  Flacco/Butterfield

Note: the basic arithmetic routines for calculating altitude, velocity, etc. not to mention the conception and original version of the program (for the KIM displays) are the work of Jim Butterfield, without whose brilliant methods of programming this would have never fit in 2 pages of memory. I am deeply indebted to JB for many of the ideas which made the graphics drivers possible, and to Eric Rehnke for helping me develop the ideas for the graphics interface.

```
Ø2[illegible] A9 3F      SINIT LDA #$ 3F     set peripheral ports
      8D Ø3 17             STA PBDD      PB=all outputs
      8D Ø2 17             STA PBD       PB=all 1's
Ø2[illegible] A2 ØB        GO LDX #$D        move 14 bytes
      Bd 49 Ø3          LP1 LDA INIT,X
      95 D4                STA BAH,X
      CA                   DEX
      1Ø F8                BPL LP1
Ø21[illegible] A2 Ø5       CALC LDX #$5        update height and velocity
Ø21[illegible] A9 Ø1       RECAL LDY #$1
      F8                   SED
      18                   CLC
Ø21[illegible] B5 D5       DIGIT LDA ALT,X
      75 D7                ADC ALT+2,X   add each digit
      95 D5                STA ALT,X
      CA                   DEX
      88                   DEY
      1Ø F6                BPL DIGIT     next digit
      B5 D8                LDA ALT+3,X   hi-order...zero...
      1Ø Ø2                BPL INCR        ...or...
      A9 99                LDA #$99
Ø2[illegible] 75 D5        INCR ADC ALT,X
      95 D5                STA ALT,X
      CA                   DEX
      1Ø E5                BPL RECAL     do next addition
      A5 D5                LDA ALT
      1Ø ØD                BPL UP        still flying?
      A9 ØØ                LDA #$ØØ      nope, turn off
      85 E1                STA DOWN
      A2 Ø2                LDX #$2
Ø23[illegible] 95 D5         DD STA ALT,X
      95 DB                STA ACC,X
      CA                   DEX
Ø23[illegible] 1Ø F9       BPL DD
Ø24[illegible] 38          UP SEC             update fuel
      A5 E9                LDA FUEL+2
      E5 DD                SBC THRUST
      85 E9                STA FUEL+2
      A2 Ø1                LDX #$1       2 more digits to go
Ø249  B5 DB          LP2 LDA FUEL,X
      E9 ØØ                SBC #$Ø
      95 DB                STA FUEL,X
      CA                   DEX
      1Ø F7                BPL LP2
      B[illegible] Ø9                BCS UPDATE    still got fuel?
Ø254  A9 ØØ        NOFUEL LDA #$Ø
      A2 Ø3                LDX #$3
Ø25[illegible] 95 DD         LP3 STA THRUST,X
      CA                   DEX
      1[illegible] FB                BPL LP3
```

```
Ø25D  A5 DD        UPDATE LDA THRUST     update thrust from joy )k
      F[illegible] 1B                BEQ THRSET    if thrust=0 motor must  off
      A9 ØØ                LDA #$ØØ        so don't update
      8D Ø1 17             STA PADD      PA=all inputs
      A9 3B                LDA #$3B
      8D Ø2 17             STA PBD       enable Y latch
      AD ØØ 17             LDA PAD       read one axis of joystick
      4A 4A                LSR/LSR A     get MSD in LSD position
      4A 4A                LSR/LSR A
      C9 Ø9                CMP #$9
      3Ø Ø2                BMI OK
      A9 Ø8                LDA #$8
Ø278  AA             OK TAX
      E8                   INX
      86 DD                STX THRUST    1 ≤ THRUST ≤ 9
Ø27C  A5 DD        THRSET LDA THRUST     set acceleration
      38                   SEC
      E9 Ø5                SBC #$5
      85 DC                STA ACC+1     acc=thrust-5
      A9 ØØ                LDA #$Ø
      E9 ØØ                SBC #$Ø
      85 DB                STA ACC
Ø289  D8           BALTCON CLD           convert ALT to hex for BALT
      A5 D5                LDA ALT            (bird altitude)
      29 ØF                AND #$F
      85 E2                STA BALT
      A5 D5                LDA ALT
      4A 4A                LSR/LSR A
      4A 4A                LSR/LSR A
      F[illegible] ØB                BEQ DBL       ALT≥ 1ØØØ?
      AA                   TAX           yes, do multiple addition
      18                   CLC
Ø29A  A9 ØA          BL1 LDA #$A         decimal 10
      65 E2                ADC BALT
      85 E2                STA BALT
      CA                   DEX
      D[illegible] F7                BNE BL1
Ø2A3  Ø6 E2          DBL ASL BALT        BALT=BALT x2
      A5 D6                LDA ALT+1
      C9 5Ø                CMP #$5Ø
      3Ø Ø2                BMI DISPLAY
      E6 E2                INC BALT      BALT= (ALTitude/50) hex
Ø2AD  A9 3F        DISPLAY LDA #$3F      draw the pictures
      8D Ø2 17             STA PBD       disable the joystick
      A9 FF                LDA #$FF
      8D Ø1 17             STA PADD      PA= all outputs
Ø2B7  A9 CD        DISBIRD LDA BIRDBAL (#$CD)   draw the bird
      85 D3                STA BAL       set the base address
      A9 14                LDA #$14      vertical positioning
      85 E3                STA RELOS
      A[illegible] 19                LDY #$19      number of points in bird
      2[illegible] 57 Ø3             JSR DISFIG    print it!
Ø2C4  A5 DD        FLAMEON LDA THRUST    do we have ignition?
      F[illegible] 17                BEQ DISPAD      not if thrust is zero
      A5 E1                LDA DOWN      are we still in the air?
      F[illegible] 13                BEQ DISPAD      not if DOWN is zero
Ø2CC  A9 E7        DISFLAME LDA FIREBAL (#$E7)    draw the flame
      85 D3                STA BAL       set the base address
      38                   SEC
      A9 1D                LDA #$1D      vertical offset
      E5 DD                SBC THRUST
      85 E3                STA RELOS     RELOS= 1D-thrust  this keeps
      A5 DD                LDA THRUST      the flame next to the bird
      ØA                   ASL A         how big should the flame be?
      A8                   TAY
      88                   DEY           Y= 2(thrust) -1  number of points
      2[illegible] 57 Ø3             JSR DISFIG    print it!
```

2

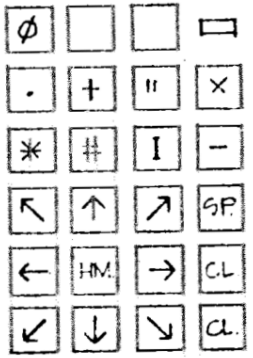
```
Ø2DF  A2 1Ø     DISPAD LDX #$1Ø      landing pad width...
Ø2E1  A9 1A        DP  LDY #$1A        and elevation
      8C ØØ 17         STY PAD        draw a line a point at a time
      CE Ø2 17         DEC PBD
      8A               TXA
      18               CLC
      69 3D            ADC #$3D       horizontal centering
      8D ØØ 17         STA PAD
      EE Ø2 17         INC PBD
      CA               DEX
      1Ø EF            BPL DP         done the pad yet?

Ø2FC  A5 D5     MOVEA LDA ALT        transfer the vital statistics
      85 E5           STA VIT+3        for display as digits
      A5 D6           LDA ALT+1
      85 EA           STA VIT+2

Ø3ØE  A5 D9     MOVEV LDA VEL+1      show velocity as absolute value
      A6 D8           LDX VEL
      1Ø Ø6           BPL MOVV
      38 F8           SEC/SED
      A9 ØØ           LDA #$Ø
      E5 D9           SEC VEL+1
      85 EC     MOVV  STA VIT+4

Ø3ØC  A5 DE     MOVEF LDA FUEL
      85 E9           STA VIT+1
      A5 DF           LDA FUEL+1
      85 EB           STA VIT+Ø
      D8              CLD

Ø315  A2 Ø4     DISNUN LDX #$4       display 5 locations
      A9 ØØ            LDA #$Ø
      85 E6            STA HOFST      horizontal offset
      AØ ØØ            LDY #$Ø        spacing flag: xx xxxx xxxx

Ø31D  B5 E8        DN1 LDA VIT,X      get a byte
      4A 4A            LSR/LSR A      get the MSD
      4A 4A            LSR/LSR A
      2Ø 79 Ø3         JSR CONVSEG    convert to segments and shine
      B5 E8            LDA VIT,X      get the same byte
      29 ØF            AND #$F        this time the LSD
      2Ø 79 Ø3         JSR CONVSEG    another digit lit
      CA               DEX
      3Ø ØE            BMI OUT
      88               DEY
      1Ø EA            BPL DN1
      18               CLC            advance the horizontal offset
      A5 E6            LDA HOFST        to space out between values
      69 14            ADC #$14
      85 E6            STA HOFST
      AØ Ø1            LDY #$1
      DØ DF            BNE DN1        unconditional branch

Ø33E  A5 E1        OUT LDA DOWN
      DØ Ø3            BNE CALJMP
      4C AF Ø2         JMP DISPLAY
Ø345  4C 12 Ø2  CALJMP JMP CALC

Ø349  Ø3 45 Ø1    INIT .BYTE 3,45,1,Ø,99,81,Ø,99,97,2,8,Ø,Ø,1
      ØØ 99 51
      ØØ 99 97
      Ø2 Ø8 ØØ
      ØØ Ø1

Ø357  B1 D3     DISFIG LDA (BAL),Y   get the coordinates
      4A 4A            LSR/LSR A      extract the Y-coord
      4A 4A            LSR/LSR A
      18               CLC
      65 E2            ADC BALT       add the bird's altitude (hex)
      65 E3            ADC RELOS      add the vertical offset
      8D ØØ 17         STA PAD        this is the Y-coord to show
      CE Ø2 17         DEC PBD        latch it in
      B1 D3            LDA (BAL),Y    get the same coordinates
      29 ØF            AND #$F        this time get X-coord
      18               CLC
      69 4Ø            ADC #$4Ø       horizontal centering
      8D ØØ 17         STA PAD        this is the X-coord to show
      EE Ø2 17         INC PBD        latch it in
      88               DEY
      1Ø DF            BPL DISFIG     done all the points yet?
      6Ø               RTS

Ø379  84 FC    CONVSEG STY TEMP      display one digit as 7 segments
      A8               TAY
      B9 E7 1F         LDA TABLE,Y    get the KIM segment code
      85 E7            STA SEGS
      86 F5            STX XREG
      A2 Ø6            LDX #$6        do seven segments

Ø385  Ø6 E7        CS1 ASL SEGS       do we do this segment?
      1Ø 35            BPL DECRX      not if bit 7 = Ø
      BD F9 Ø3         LDA SEGTBL,X   find out where the 5 dots for
      29 ØF            AND #$F          each segment start
      85 ED            STA VPOS       first in the vertical
      BD F9 Ø3         LDA SEGTBL,X
      4A 4A            LSR/LSR A
      4A 4A            LSR/LSR A
      29 Ø7            AND #$7        then the horizontal
      18               CLC
      65 E6            ADC HOFST      this is where the digit is
      85 EE            STA HPOS         in the row of digits
      AØ Ø4            LDY #$4        do 5 dots per segment
Ø3AØ  A5 ED       DISPT LDA VPOS
      8D ØØ 17         STA PAD
      CE Ø2 17         DEC PBD        latch the Y-cord.
      A5 EE            LDA HPOS
      8D ØØ 17         STA PAD
      EE Ø2 17         INC PBD        latch the X-cord.
      BD F9 Ø3         LDA SEGTBL,X   is it to be up-and-down...
      3Ø Ø4            BMI HL           ...or side-to-side?
      C6 ED            DEC VPOS
      1Ø Ø2            BPL DECRY      unconditional branch
Ø3B9  C6 EE          HL DEC HPOS
Ø3BB  88          DECRY DEY           done 5 dots?
      1Ø E2            BPL DISPT
Ø3BE  CA          DECRX DEX           done 7 segments?
      1Ø C4            BPL CS1
      A6 F5            LDX XREG
      A5 E6            LDA HOFST      advance to the next digit place
      18               CLC
      69 ØC            ADC #$C
      85 E6            STA HOFST
      A4 FC            LDY TEMP
      6Ø               RTS

Ø3CD                  F4 F5 F6       BIRDBAL = Ø3DC
Ø3DØ  E3 E7 D2 D8 C1 C9 B1 B9        FLMBAL  = Ø3E7
Ø3D8  A1 A2 A8 A9 9Ø 93 94 95        SEGTBL  = Ø3F9
Ø3EØ  96 97 9A 8Ø 8A 7Ø 7A Ø5
Ø3E8  Ø5 15 15 25 25 34 36 44
Ø3FØ  46 54 56 64 66 73 77 83
Ø3F8  87 DC EB 65 DØ Ø5 ØB D6

**********
```

# TVT-6 etch-a-sketch

by Michael Allen
6025 Kimbark
Chicago, Ill. 60637

This program illustrates one way to overcome one of TVT-6's limitations, a snowy screen during program execution, which would seem to rule out animated displays. The sketch program is entered by a subroutine jump inserted in your TVT-6 scan program at address 17C9 (assuming the scan program begins at addr. 1780). As long as not too much time is taken away from scan the screen image stays fairly stable.

Load the sketch program, and scan program (set addr. 17C9 to 20 00 00). Start at addr. 17AD, and your display should be filled with Ø's. The Kim-1 keyboard will now function as follows:




The arrows indicate the direction of cursor travel when the key is depressed. Keys 3 and 7 clear the screen. Keys B through DA determine the character trail left by the motion of the cursor. Key B will leave a trail of blanks. Keys +, GO, and PC, fill the display with one character. Key 5 homes the cursor to center screen.

If you have added a keyboard to Kim with a different arrangement of keys, simply change the values in the table at addr. 009B.. These can also be changed for different character trails.

For the effect of animated motion, delete the key debounce option by inserting NOP's at addr. 0025 through 0029.

I found that I could not live with the Kim-1, TVT-6 combination for long without more memory. So I have added S.D.Sales 4K board as per Bob Haas' article in the April '77 Kilobaud. By changing the jumpers from Kim's on board memory to the appropriate points on the new board (and restoring Kim's cut foil trace), and by changing the scan program locations 17AA to 88, and 17D2 to 86; memory pages 0E and 0F will be displayed.

I will send along two programs for Kim-1, TVT-6 with added memory as soon as I type them up. (Sure wish I had a printer!) One is "Life" (takes less than a second per generation), and the other is "Pong" (uses Kim's keyboard to move the paddles).

```
XXXXXXXXXXXXXXXXX  KIM-1 - TVT-6 SKETCH PROGRAM  XXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXX  INPUT AND OUTPUT  XXXXXXXXXXXXXXXXXXXXX

0000  48              PHA         SAVE ...
0001  8A              TXA               SCAN ...
0002  48              PHA                    REGISTERS.
0003  20 40 1F        JSR KEYIN   KEY PRESSED?
0006  F0 1D           BEQ OUT     NO; BACK TO SCAN.
0008  20 6A 1F        JSR GETKEY  YES; GET KEY CODE.
000B  AA              TAX         USE KEY AS INDEX TO TABLE.
000C  C9 0B           CMP #$0B    IS IT 0 TO A?
000E  90 1E           BCC NUMB
0010  C9 12           CMP #$12    OR B TO AD?
0012  90 2A           BCC LETT
0014  B5 9B           LDA TABLE,X MUST BE +, GO, OR PC.
0016  D0 02           BNE NOCLR   FORCE BRANCH AROUND CLEAR.
0018  A9 20  CLEAR    LDA #$20    ASCII BLANK.
001A  A2 00  NOCLR    LDX #0
001C  9D 00 02  LOOP  STA DISP-1,X FILL DISPLAY ...
001F  9D 00 03        STA DISP-2,X          WITH CHARACTER.
0022  E8              INX         NEXT ...
0023  D0 F7           BNE LOOP
0025  20 FE 1E  OUT   JSR AK      OPTIONAL KEY
0028  D0 FB           BNE OUT     DEBOUNCE.
002A  68              PLA         RETORE ...
002B  AA              TAX               SCAN ...
002C  68              PLA                    REGISTERS.
002D  60              RTS         RETURN TO SCAN.

XXXXXXXXXXXXXXXXXXXXXXXX  MAIN PROGRAM  XXXXXXXXXXXXXXXXXXXXX

002E  B5 9B     NUMB  LDA TABLE,X GET SUBROUTINE ADDRESS.
0030  85 33           STA SBR XX  STORE IT.
0032  20 XX 00        JSR 00XX    JUMP TO IT.
0035  A2 00           LDX #$XX    ADDR. 0036 = "CHARPO".
0037  A9 20           LDA #$XX    ADDR. 0038 = "CHARAC".
0039  9D 00 02        STA DISP,X  ADDR. 003A,3B = "LINE","PAGE".
003C  D0 E7           BNE OUT     FORCE BRANCH OUT.

003E  B5 9B     LETT  LDA TABLE,X GET NEW ASCII CHARACTER.
0040  85 38           STA CHARPO  STORE IT.
0042  D0 E1           BNE OUT     FORCE BRANCH OUT.

0044  A9 0F     HOME  LDA #$0F    SET CHARACTER POSITION ...
0046  85 36           STA CHARPO       TO CENTER ...
0048  A9 E0           LDA #$E0               OF SCREEN.
004A  85 3A           STA LINE    (OR ...
004C  A9 02           LDA #2            THEREABOUTS ...
004E  85 3B           STA PAGE                    AT 02EF)
0050  60              RTS

XXXXXXXXXXXXXXXXXXXX  DIRECTION SUBROUTINES  XXXXXXXXXXXXXXXXXXX

0051  20 7F 00  LEFTUP JSR LEFT     LEFT FIRST,
0054  38        UP    SEC          THEN UP. PREPARE TO SUBTRACT.
0055  A5 3A           LDA LINE
0057  E9 20           SBC #$20     MOVE UP A LINE.
0059  C9 E0           CMP #$E0     OFF TOP OF PAGE?
005B  D0 08           BNE ENTER1   NO; ENTER NEW VALUE.
005D  A2 02           LDX #2       YES;
005F  E4 3B           CPX PAGE     OFF TOP OF DISPLAY?
0061  F0 04           BEQ RTN1     YES; RETURN.
0063  C6 3B           DEC PAGE     NO; MOVE UP A PAGE.
0065  85 3A     ENTER1 STA LINE
0067  60        RTN1  RTS

0068  20 7F 00  LEFTDN JSR LEFT     LEFT FIRST,
006B  18        DOWN  CLC          THEN DOWN. PREPARE TO ADD.
006C  A9 20           LDA #$20     MOVE ...
006E  65 3A           ADC LINE           DOWN A LINE.
0070  C9 00           CMP #0       OFF BOTTOM OF PAGE?
0072  D0 08           BNE ENTER2   NO; ENTER NEW VALUE.
0074  A2 03           LDX #3       YES; IS IT OFF BOTTOM ...
0076  E4 3B           CPX PAGE           OF DISPLAY?
0078  F0 04           BEQ RTN2     YES; RETURN.
007A  E6 3B           INC PAGE     NO; MOVE DOWN A PAGE.
007C  85 3A     ENTER2 STA LINE
007E  60        RTN2  RTS

007F  A5 36     LEFT  LDA CHARPO   AT LEFT EDGE OF SCREEN?
0081  F0 02           BEQ RTN3     YES; RETURN.
0083  C6 36           DEC CHARPO   NO; MOVE LEFT.
0085  60        RTN3  RTS

0086  A9 1F     RIGHT LDA #$1F     AT RIGHT EDGE ...
0088  C5 36           CMP CHARPO             OF SCREEN?
008A  F0 02           BEQ RTN4     YES; RETURN.
008C  E6 36           INC CHARPO   NO; MOVE RIGHT.
008E  60        RTN4  RTS

008F  20 86 00  RIGHTUP JSR RIGHT  RIGHT FIRST,
0092  4C 54 00        JMP UP       THEN UP.

0095  20 86 00  RIGHTDN JSR RIGHT  RIGHT FIRST,
0098  4C 6B 00        JMP DOWN     THEN DOWN.
```

XXXXXXXXXXXXXXXXX TABLE OF SUBROUTINE ADDRESSES XXXXXXXXXXXXXXXXX

0098 58 66 95 18 73 44 85 13 54 85 20 2A 23 43 ← KEY FUNCTION
↙ → ↘ CL ← HM → CL ↖ ↑ ↗ SP • # I

00AA 2D 2E 2B 58 4F 22
\- . + X Ø "

4

# TVT6

Philip K. Hooper  Box 293, Johnson, VT  05656

## SOME CHEAP, EASY, and HELPFUL TVT-6 HARDWARE MODIFICATIONS

1. Replace resistor R9 with a 5 Megohm pot. This permits varying the cursor 'blink rate' from a slow cycle of several seconds per blink up to a rate fast enough so that the cursor appears to be on continuously.

2. From the junction of R19 and D5 (see diagram), connect:
   a. one diode to the jumper parallel to R19 (connects to pin 15 on the 2513)
   b. one diode to the long jumper running beneath the 2513 (connects to pin 16 of 2513)
   c. one 1K resistor

   Connect the other end of this 1K resistor to:
   a. pins 11 and 12 of the 74165 shift register (remove chip, bend pins up, replace chip, wrap or carefully solder to uplifted pins)
   b. a parallel combination of a 3K resistor and a .01 capacitor going to ground (the jumper immediately 'beneath' the 74165 is a convenient ground line)

   This modification changes the cursor from a glob which overwhelms the character it tops, into an UNDERLINE which extends two dots to the right of the indicated character and, hence, remains discernible even when used with the character 'E'. It may also be used to draw a solid horizontal line.

1 x 44, double-size character TVT-6 driver subroutine, Screen-centered.
Displays $0414 - $083F.

```
1780 8D8517    insert SCANNI address          * Byte in 179F determines page,
1783 2034BC    character line scan                80, 81, 82, 83
1786 E684      increase 'half-a-row'          * Byte in 1784 determines words,
1788 C9C0      done?                              14, 54, 94, D4
178A D0F4  (1780)  NO
178C A9        set X00 to 00                  horizontal scanning time - 65 microseconds
178D A211      blank row count to X               239 blank lines
178F E683      increase frame count by 1 (carry IS set)   12 active lines
1791 F015      (17A8) DONE2                      256 total lines
1793 2034BC    first block                    16640 microseconds for these lines
1796 E8          of blank                       28 microseconds for V Synch.
1797 D0FA      (1793)   scans                 16668 microseconds/frame.
1799 2034C0    V.Synch                        Vertical Frequency: 59.9952 Hz.
179C 8583      stash frame count
179E A900      reset SCANNI address, row counter (bits 3-5)
17A0 2034BC      second block
17A3 E8          of blank
17A4 D0FA      (17A0) scans
17A6 F0E9      (1791) do another frame
17A8 A583      reset frame counter
17AA 60        get out
17AB

                                              For single-height characters:
                                              [1787] = 08    [178E] = 09
                                              (8 more blank scans to fill in for
                                               the 8 'lost' active scan lines)
```

To use as a 'lock-in' routine, without subroutine return, merely change the byte at 17AB to 4C, putting more garbage in 'wastebasket Y'. In addition, 179C may be changed to 84 to suppress occasional 'flashes'.

```
1780 8D8517
1783 2034BC C604
1786 C9C0 98F4
1788 84 A211
178D 85 83 F015
178F 2034BC E8
1792 D0FA F0D8
1794 A583 60
1796
```

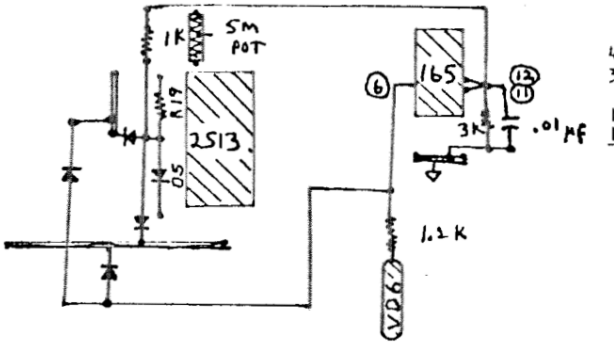
The blank lines are scanned in two separate blocks around the V. Synch pulse to put the actively scanned line in the center of the screen instead of at the bottom.

The Program is entered with the timing parameter in the accumulator, followed by a JSR to 178D.
e.g. A9C0  208D17 . . .

Since the frame counter is incremented, low values of the timing parameter produce the longest reside times, while a large value (like F0) permit only a short stay (16 frames, about [redacted] 1/4 second) in the routine.

3. Connect a 1.2K ohm resistor to the (otherwise-unused) edge finger VD6. From the other side of this resistor, run:
   a. a wire to pin 6 of the 74165 (hope you lifted it already for cursor modification)
   b. two diodes, which go to the same two jumpers as did the new cursor diodes.

   This modification results in a small 'lump' appearing at the lower left corner of any character having bit 6 HI. (the lump is 1 dot wide by 2 dots high). In this way we gain a sort of pseudo-upper-case and, along with the cursor modification, are able to distinguish between 256 different characters - that is, we can now determine the complete bit pattern of a byte from its image on the screen.




Components Required:

4 small signal diodes
3 1/4 W resistors
  1K, 1.2K, 3K
1 0.01 mfd capacitor
1 5 Megohm potentiometer

(These values were arrived at by 'cut and try' and, although they work for my rig, they can most likely be improved upon by someone with hardware expertise. I would appreciate hearing from anyone who knows what the values 'should be'.)

De[illegible]

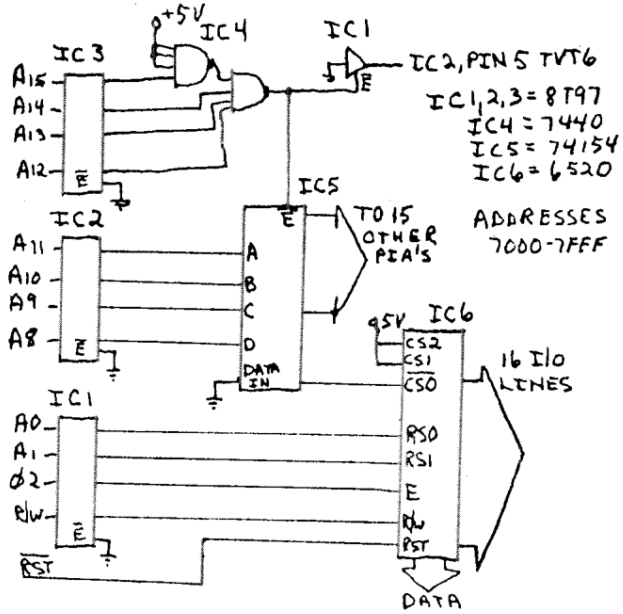
[illegible] enclosed one possible configuration of expansion decoding. It is specifically de[illegible] th TVT6 in mind (TVT6 from Popular Electronics). KIM will operate normally as with ju[illegible] the TVT6.

[illegible] responds to addresses 7000-7FFF. Each port or section is one page wide. Currently, I am using each section for an IN/OUT port.

IC1's output attaches to IC2, pin 5 of the TVT6. This will disable normal KIM operation when low. IC2, pin 5 (TVT6) will float high when 7000-7FFF is not selected. The two high enables (CS1 and CS2) on the 6520's go to five volts and the outputs of the 74154 go to the active low chip select (CS3) of the 6520's. Note that the data in the pin of the 74154 goes to ground. It could just as easily be tied high for an active high signal out.

The decoding is not down to every single address but still allows for 20k of expansion between 2000-6FFF. Achieving low parts count and later decoding freedom was the purpose of this design. This circuit plus data buffers and two 6520's will fit on one Radio Shack 4½ X 4 inch board.

I am considering a second processor to drive the TVT6 transparently to free KIM for normal use (an intelligent terminal?). I would like to hear from others thinking along similar lines.




D. Clem
RR #2,
Spencerville, Ohio 45887

TVT-6 Remarks by Cass and Dan Lewart 12 Georjean dr., Holmdel, NJ 07733

This ingenious and simple KIM/TV interface was described by Don Lancaster in Popular Electronics (July/August 1977) and in Kilobaud (Dec. 77/Jan. 78). The complete kit (without the 36-pin connector) is being sold by PAIA Electronics, Box 14359, Oklahoma City, OK 73114 for $34.95. Here are some observations based on our experiences building and experimenting with it. If you have any hardware questions write to Cass, and send software questions to Dan.

1. The kit is easy to build (2 hours) but connections to KIM require a neat soldering job (4 hours).

2. All connections between the TVT and KIM are between the TVT socket, the KIM expansion connector and the KIM board. You can avoid making any connections to the KIM Application connector by breaking the foil to the A-K pin.

3. If you decide to convert your TV set into a monitor use the base of the first video amplifier as your input and increase the emitter resistor of this stage until the ASCII characters are steady and not leaning.

4. If the right sides of all ASCII characters are missing, lower the value of C5 to 68 pF and replace R11 with a 500 ohm potentiometer.

5. The following refers to the 16 x 32 character program supplied with the kit and the only one we successfully used so far:

   It is possible to display from 4 lines (½ page) to 18 lines (2¼ pages) at a time. Unfortunately, the display always ends at the top of a page. The following locations control the memory area to be displayed:

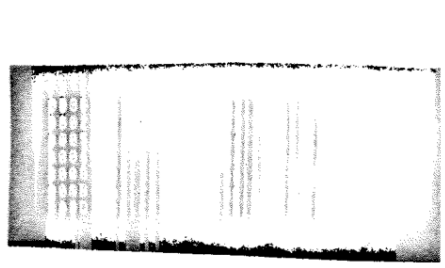
| Location | Contents | Bit Pattern |
|---|---|---|
| 17AA | MSB of first address after last displayed line "OR"ed with 80 | 10000xxx |
| 17CD | LSB of first address to be displayed has to be a multiple of 20 | xxx00000 |
| 17D2 | MSB of first address to be displayed "OR"ed with 80 | 100000xx |

   E.g. to display 0200-02FF: 17AA=83, 17CD=00, 17D2=82. You may have to adjust the vertical hold to keep the picture steady. Displaying page 0 you will see the important locations EF-FF. To display most of page 1 move the stack pointer to a lower address, preferably 1F (LDX $1F,TXS) so the stack still fits.

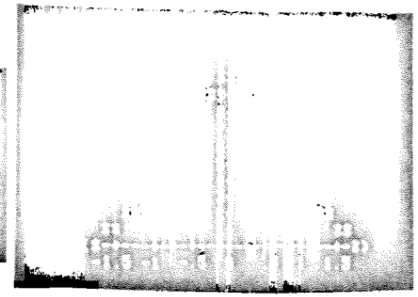
6. You start the display by JMP 17AD. To exit the display mode use the NMI interrupt by storing the location of your driver program in 17FA/17FB and by pressing "ST" to exit the display program and to start execution of your program. To get a rough frame around the display start at 37AD instead of 17AD.

7. We have written several programs for TVT-6; a Disassembler displaying 14 formatted statements at a time and checking for correct op-codes, Morse code teacher displaying the transmitted sequence of characters, and a demo program. These three programs incl. cassette and a complete description are being marketed by PAIA for $4.95.

8. The next project is to add 1 K of RAM to our KIM by piggy-backing eight 2112-2s and to store the display and monitoring programs in that upper K. Will report on success (or failure).



Disassembler



Demonstration Program

Pictures taken off the TV screen

**********

Ronald Kushnier
9108 Addison Court
Cornwells Hts., Pa. 19020

## NOTES ON THE TVT-6

Now that the Master Merlin (Don Lancaster) has returned to his retreat somewhere in Arizona (maybe someday he'll publish his address), it appears that it's up to us common folk to continue the magic of the TVT-6.

Several items which were glanced over in the construction articles become very apparent when actually using the interface.

1. Memory Expansion

   The TVT-6L - lower case board is set up to use memory locations 2000 on-up, so that KIM expansion is limited to the lower 4K option.

   The TVT-6 - upper case only board is set up to use memory locations 8000 on-up, so that somewhat more memory can be included with, of course, additional decoding.

   What this means is that you should carefully choose your system requirements before you choose your board. PAIA has admitted problems with the TVT-6L boards and is making its big push with the upper case only board.

2. THE TVT-6/KIM Terminal

   The Full Performance Cursor Program works great* although I'm still trying to figure out what a "Spare Hook" is. The software does turn KIM into a terminal. However, once you get the KIM up and running with this program, the thought that crosses your mind is "Gee, I wish I had a computer to hook up to this fine new terminal". To get KIM to be interactive, as both terminal and computer is a whole different ball game. I am now investigating the possibility of using a hardware interface as a UART hooked through KIM's serial port. This would make possible the use of KIM's serial interface firmware. However, this approach may be a case of the dog trying to chase it owns tail.

| *NOTE: | Instruction 0185 should be 03 instead of 01 to obtain proper scrolling. Also, individual control codes can be changed to accomodate different keyboards. (See Radio Shack keyboard hook up.) |
|---|---|

3. A Little Word Called Interrupt

   A problem which immediately becomes apparent is that the SCAN routine is a trap. Once you're in it the only way out is through an interrupt. It would have been nice if SCAN has been a subroutine like KIM's SCAND that you could jump to whenever you wanted to display something, but the SCAN timing is critical and I have had little success in modifying that program.

   So, up to this point, the only way I have found for KIM to continually update the display on it's own is to use the interval timers in the interrupt mode.

4. More Memory (SLURP!)

   Using the TVT-6 gives you an insatiable appetite for more memory. Until I see a SCAN program for displaying just part of one page, I am forced to use 2 pages for display. That doesn't leave much room for an applications program or word storage.

   Another funny thing happens when you go video- you don't want to look at the seven segment readouts any more. They become totally passe. This must be caused by some psychological factor like watching TV for all these years.

I am hoping the great Merlin will reappear soon! Until then, I would like to correspond with anyone using the TVT-6.

## ASSEMBLING THE TVT-6

One of the many reasons why I went to PC 77 at Atlantic City was to tell PAIA Electronics what I thought of them. After all I had ordered Don Lancaster's TVT-6LK Kim/Video Interface right after his original article came out in Kilobaud in May (June 1977 issue). And it was now the end of August and still I had heard nothing! Well, PAIA was at the Convention and they told me about late deliveries and production problems etc, etc. Anyhow, I purchased a PVI-1K, which was equivalent to the TVT-6 appearing in the July and August issues of Popular Electronics. PAIA had a working unit on display and it looked great. They had taken Don's KIM connections literally and had used the expansion connector for the internal KIM/Video Interface. I had determined from the very start that this approach was unacceptable and that I would not sacrifice my expansion capabilities.

### KIM Expansion Rationale

I have had the basic KIM for a year now, and if anyone is worried that they will not have enough to do with a personal computer, my wife will testify to the fact that it has been a continual hassle to pull me away from the unit night after night after night. KIM has limitless applications. Over the time, however, I have had the urge to expand. The question I ask myself is "What can I expect from a fully expanded system?" The answer is a system with a decent Basic operating program, and video and cassette interface. Now, by buying an adaptive mother board, additional power supplies, memory, a video board and so forth, KIM could be expanded to provide any desired system. This would take several hundred dollars. With "PET" just around the corner, this piece-part approach makes little sense to me. Therefore, I decided to keep KIM as simple as possible with expansion limited to as low a dollar figure as could be achieved. This approach included a Radio Shack ASCII Keyboard Kit (I already had the IC's), the TVT-6 video interface and eventually a low power 4K memory board, which would simply plug into the KIM expansion connector. I originally

was ing to use a personal portable TV (A gift for my wif s a display, but I picked up a surplus monitor for $45.00 from Selectronics, 1201-25 So. Napa Street, Phila., Penna. 19146.

The screen was a little discolored from ten years of constant use, but who cared. After inserting the two required parts ( a capacitor and width coil) she ran fine. So this was going to be my expanded system. At less than $100 invested (minus the memory), I figured it would hold me for a while.

Building the PVI-1K

The PVI-1K Kit was somewhat disheartening, the first problem was the 36 pin mating connector. It did not come with the kit. The 'Pop' Tronics article stated the Kit contained "all of the above parts" and one of those parts was the connector. A call to PAIA resulted in frustration. I couldn't get past the receptionist. "Yes, it was advertised, but we are not supplying any; and I don't know why", was the terse reply. I did finally manage to scrounge up a 72 pin version, but it was not easy to come by.

The advertisement said "sockets" and a strip of Molex Solder Cons were supplied. Well, I guess some people would call them sockets, but I wouldn't use them. To me, it was worth a couple of extra bucks for the real thing. When installing the sockets, I noticed that the registration of the PC board was far from perfect. Several of the holes were not exactly where they should have been and a few had not been totally drilled through.

All the land on the PC board was unprotected copper. This corrodes fairly fast so I would advise cleaning with Scotch Brite before fabrication. I tinned all the land including the edge connector lands during assembly. This provided a less corrosive finish. A small amount of liquid flux applied to the patterns made the job easy. The excess flux is easily removed with alcohol when finished.

The board went together easily. There were no other surprises.* I installed miniature spdt switches for the cursor and line length jumpers. These switches were obtained from Poly-Paks. A dpdt switch for conversion back and forth from KIM to TVT was mounted using epoxy ribbon on one of the brackets needed to mount the card connector. These brackets, by the way, were made from sawed off card pullers.

* Except C5 was changed from 2200 pF to 240 pF to get the timing right.

When I tried to read in the PAIA/KIM cassette, I found the record level was too low for the KIM to respond- so back it went to PAIA.

KIM Modification

Since I refused to give up the expansion connector to the video interface, I needed a new insertion point for the numerous inter-connections required for the TVT-board. I struck at the heart of KIM - the 6502. Here were most of the points I needed, and it was close to the new 36 pin mating connector which I installed at the top of the KIM board. I knew I would have to be extremely careful when "operating" in this area. It was an "all or nothing" operation, but I decided to go ahead.

The first thing I did was to make a Xerox of the bottom of the KIM board. This technique is surprisingly effective. I have used it several times before on other projects to make templates for drilling. The Xerox detail is remarkably clear and useful. With this picture of KIM's bottom, I was able to draw in exactly where the new wires would be placed. Some special tools I needed were the Vector Wiring Pencil, liquid flux, a precision tweezers, epoxy ribbon and a three wire grounded soldering iron. With my trusted wiring pencil in hand, I proceeded with the operation. It was not easy. When your're working with wire not much thicker than a human hair, things get a little tedious. By applying a tiny dab of liquid flux on each connection, things were made somewhat better. Also, the insulation was burned off the wire and it was properly tinned before applying it to the land to be soldered. The fine wires were held to the board with small dots of epoxy ribbon putty at strategic points. The modification was slow and painstaking, but when finished did not look too bad.

The TVT-6 provides a good, low cost expansion of your KIM's capabilities. I would not recommend my approach to a hardware novice, but if you do have some hardware and building experience by all means - go to it!

USING THE TVT-6 WITH THE RADIO SHACK KEYBOARD

The following list represents my implementation of the Radio Shack keyboard to the TVT-6 Full Performance Cursor Program. I used the NMI input to KIM instead of the IRQ input with the strobe ST. One correction to the published software C185 should be 03 instead of 01 to obtain proper scrolling.

| Function | Key | ASC II | Change in Program | | |
|---|---|---|---|---|---|
| | | | Address | From | To |
| CLEAR | CLEAR | 02 | 011B | 16 | 02 |
| CARRIAGE RTN | SHIFTED ] | 0d | --- | --- | -- |
| CURSOR UP | SHIFTED [ | 0b | --- | --- | -- |
| CURSOR DOWN | LINE FEED | 0A | --- | --- | -- |
| CURSOR LEFT | BACK SPACE | 08 | --- | --- | -- |
| CURSOR HOME | CTRL | 01 | --- | --- | -- |
| SCROLL UP | R. BLANK | 05 | 0137 | 11 | 05 |
| SPARE HOOK | BREAK | 00 | 013B | 12 | 00 |
| ERASE TO END | HERE IS | 03 | 013F | 13 | 03 |
| CURSOR RIGHT | TAB | 09 | ---- | -- | -- |

*

The published program is designed for wrap around scrolling. For use as open ended scroll change 0147 from 20 (C2) (01) to 4C 75 01.

* See Popular Electronics August 1977

Ronald Kushnier
3108 Addison Court
Cornwells Heights, Pa. 19020

**********

This is not elegant. It isn't even quick and dirty. Slow and dirty is about the best I can offer, but it works. I'm still trying to figure out how to operate the TVT-6. I eliminated the vertical blanking portion of Table II and used that interval (tracked by the timer and interrupt) for processing.

CHANGES TO TAPLE II IN THE TVT-6 ARTICLE

```
17AD  A9 8D     INTOUT  LDA  #8D     Load timer for interrupt
17AF  8D 0C 57          STA  CLK1I   plus free Vertical sync
17B2  68                PLA          Recover registers
17B3  A8                TAY          Y
17B4  68                PLA
17B5  AA                TAX          X
17B6  68                PLA          A
17B7  40                RTI          Return
                                     17B8 - 17BE not used
17BF  48        INTIN   PHA          Interrupt entry.  Save A
17C0  8A                TXA
17C1  48                PHA          X
17C2  98                TYA          and
17C3  48                PHA          Y
```

Just connect PB7 to $\overline{IRQ}$ or $\overline{NMI}$ and set that vector to 17BF. Start up with the following (relocatable) short patch and away you go.

```
0100  58        PATCH   CLI          Needed if you use IRQ
0101  A9 80              LDA  #80     Set PB7 to output
0103  8D 03 17           STA  PBDD    to allow interrupt
0106  A9 8D              LDA  #8D     Start up
0108  8D 0C 17           STA  CLK1I   interval timer with interrupt
010B  4C 00 02           JMP  PSTART  Go to program start
```

I used $8D_{16}$ cycles. This allowed my Vertical hold to be nearly normal. Increasing the number will give more instructions per scan and vice versa.

Extra: If you only have the basic KIM, changing 17AA of Table II to 85, along with a slight adjustment to Vertical hold will display pages 02, 03 and 00 consecutively allowing to fill the whole screen. In other words, a 24 line by 32 character display.

Michael Brachman
50-1 Westbrook Hills Dr.
Syracuse, N.Y. 13215



**********

....an exerpt from a letter from:
Christopher A. Harris, 507 Dabney Hall, Univ. of Cincinnati, Cincinnati, OH 45221

"...I have stumbled upon a dismaying problem: I have always wanted a video display such as the TVT-6. It appears to me that I would not be able to use such a dedicated display due to the fact that it ties up so many pins on the expansion connector and so may memory locations ($2000-$EFFF according to the First Book of Kim) Do you know anything about this?..."

Chris.

There was some confusion concerning the addressing requirements of the TVT-6 since Lancaster also introduced the TVT-6L at about the same time. As it turns out, the TVT-6 needs $8000 on up while the TVT-6L uses $2000 on up. So you can add some memory expansion to Kim.if you use the TVT-6.

**********

# FOCAL

FOCAL has been available for the 6502 for quite awhile now and offers some advantages that make it an attractive alternative to BASIC. The fact that an assembly-listing is available makes it especially beneficial to those of us who are interested in delving into the inner workings of a high-level language and perhaps modify it and/or extend to suit our whims. FOCAL includes provisions for adding to the command language and makes interfacing to machine language functions a piece of cake. BASIC offers none of this.

FOCAL is available from two sources at this time; ARESCO (P.O. Box 43, Audubon, Pa 19407) and 6502 PROGRAM EXCHANGE (2920 Moana, Reno, NV 89509). They both offer FOCAL for about the same price, however the Program Exchange has developed a library of FOCAL programs including StarTrek, so I would highly recommend that you get their flyer and see whats available (I think it costs $$$). Also they have an excellent 104 page user manual which is available for $12.00. I just received it in time to mention it in this issue and can recommend it as an effective means for becoming familiarized with FOCAL operations.

Up to this point, the biggest single disadvantage of FOCAL has been that there was no built-in way of saving and loading FOCAL programs using cassette or disc. Well, I have found a way to accomplish this and if you'll be patient I'll impart the knowledge to you.....(by the way, the absolute memory locations hold true only for the Version 3D (and possibly FCL-65E) other implementations will have to know where their particular pointers are).....

SIMPLE!!!All you have to do is to save the pointers PBADR ($31,32) and VARBEG ($3E,3F) and the data that is referenced indirectly between them. For instance: PBADR points to $360A and VARBEG points to $390F. Your storage device driver program should dump all data from $360A to $390F and also the pointers themselves which must be reinitialized when you re-load that particular program. How else is FOCAL supposed to know where that program is???

No, I haven't actually written a caasette driver for FOCAL (I use disc) but don't see any problem at all doing just that.. But, wait a minute...before we all go off on our own and write our own version of the ultimate FOCAL cassette handler, let's figure out some sort of a "standard". I thing it's important to be able to work with named records instead of our regular ID number. All we really need to do is extend the ID portion of the KIM cassette format to include a fixed number of ASCII characters (say 8) and include an area for the pointer information that we need. It's necessary that we have some proposals by the next issue so we can get started on our driver software. As far as the command extension to FOCAL is concerned, let's reserve the letters "K" for KEEP (which will save the program on cassette) and "L" for LOAD (which will load a program from cassette into memory).

We may want to use a binary recording format for increased speed and could probably "lift" some of the code from the cassette driver presented in issue #7/8 (written by John Oliver).

More next time. Got any ideas about FOCAL that you'd like to share?

**********

ear Eric:

I've had a KIM now for about two years and have enjoyed and cursed. Also have two TIM's not yet implemented and a PET 8K, so have had some experience with almost all 6502 stuff (even played with an Apple once). For what it's worth, here are some comments in random order:

(1) a lot of tape player problems are no doubt due to the fact that the output replicates the input, i.e., a signal being read is also present on the high or low output lines. This can, no doubt, in some tape players, cause all kinds of havoc--simple fix--when reading, unplug the mic or aux. Consult the KIM manual and you'll see the problem.

(2) have a KIMSI board, full of connectors, and 24K of Godbout Econoram, all of which ran when plugged in first time—no fixes, no glitches, just good results (also had a Godbout termination board). Also mounted up is a Burr-Brown 16 chan A/D which is expensive for home hackers but works well.

(3) Terminal is a XITEX kit with CBC monitor—no troubles with the kit other than the video out looks impossible on a color TV and horizontal lines are more intense than vertical ones--could be annoying.

(4) So much for hardware--I must say I've treated the KIM board shabbily like pulling off keyboard and displays, messing up for TVT-6, etc.--and it still works.

(5) Yes, I tried TVT-6 and that too worked pretty well, BUT the display drops out if you are computing which is annoying to say the least. Cheap thrills for the home hacker and very useful for that but not for serious business.

All of which brings us to software--I have two languages up and running—FOCAL from the 6502 Program Exchange and Microsoft Basic via Johnson Computer. I'll try to remain objective and describe what's going on. First, I'd better explain that this system was supposed to be a desktop computer and data acquisition system, and so my requiremonts, especially on software, are somewhat more stringent than the average hacker's might be.

The first package I acquired was the Microsoft Basic. Put it on the recorder, wouldn't read in. Tried several other tape recorders. Finally found one that would read 2 out of 3 times (after diddling with the head alignment). Beware--recorders need good high frequency response for hypertape. Some can't deliver. Ordered 2 extra copies of the tapes, same problem. Sent them all back, and Johnson Computer verified them all and fixed some bugs in the process. This reading problem is bothersome but cannot really be blamed on anyone in particular--just think of the quality of some of the components we're using! Another, more serious problem with Microsoft Basic is that if it hangs up, for example, in a bad Read operation, or if for any reason you want to get back to the KIM monitor, the Basic crashes on reset and has to be reloaded. I've had some conversations (yes, plural) with Johnson Computer about this with no result. They can't help an awful lot anyway because they don't have a source listing from which to work, and I haven't time for a lot of blind poking around to provide a fix.

In the instructions, there is a letter from Microsoft which says, "...feel free to give us a call..." You can, but you won't be allowed to talk to anyone helpful, and will be referred back to Johnson Computer. Catch-22. As of this writing, no help is forthcoming.

The FCL-65E from the Program Exchange was, on the other hand, fully supported with a users manual, two cassette tapes, and a complete source listing with instructions for hackers and even memory allocation and calling routines for hacking built into the interpreter. This language read in first time on my machinery with no problems whatever. Easy to get in and out to KIM by reset and you can diddle with the language to your heart's content. FCL-65E does, however, have its drawhacks for KIM. There is no provision for cassette I-O even for programs; it will have to be written. The present version is slow. For those who have grown up with BASIC or FORTRAN, FOCAL will be a little strange, but it is much more flexible and compact than BASIC. There are no built-in routines for trig functions, log, or exponential but some written in FOCAL are suggested; I intend to try an arithmetic chip like National Semiconductor's.

I guess what I'm trying to say is that if you are content to use a language as it is, the Microsoft Basic is OK, even good, but you w be able to do much effective hacking due to lack of source listing or support services. If you're a dyed-in-the-wool hacker, FCL-65E is a far superior purchase. A language without the source listing is useless to me; I won't buy another, which no doubt severely restricts my choices but I'll have to put up with it. I'm looking forward to 6502 PASCAL. if and when.

With regard to PET, not too much to say. It's a good machine, but I've been bombarded with proposals from Commodore to buy a bunch of very expensive hardware and software but after 8 months, don't yet have an operating manual or a de-bugged ROM; some of their priorities seem a little out of whack.

On balance, I'm enjoying my turbulent affair with microcomputing; the education, although sometimes frustrating, has been mostly fun. Keep up the good work.

Sincerely,

DON L. LATHAM
Research Meteorologist/Physicist

**********

BOOK REVIEW by the editor

THE CHEAP VIDEO COOKBOOK by Don Lancaster

Lancaster has done it again with his latest effort. This book is all about the ins & outs of low cost video interfacing (you never would have guessed, right?).

The first half of this 250 page book is devoted to software and hardware design techniques for video displays. Lancaster's approach is a software-intensive one using the minimum necessary hardware.

(The same state-of-the-art principles which led to the development of KIM).

If you have already read his previous work "TV Typewriter Cookbook", you would be well on the way to getting the most out of "The Cheap Video Cookbook". If you haven't read it - then I suggest you do-before you tackle Lancasters latest. (beginners take note).

The rest of the book delves into a new - and even more devious TVT - the TVT-6 5/8.

in the words of the author-

"...This is a third generation design that picks up the best features of the TVT-6 and TVT-6L that earlier appeared in various issues of Kilobaud and Popular Electronics. New features added include the full graphics ability, transparency options, a simpler and cheaper overall circuit, and much more modest use of microcomputer address space..."

I strongly recommend you purchase this book, and his previous one, if you are interested in the use of his low-cost TVT design in your system.

"The Cheap Video Cookbook" deserves careful study by all students of advanced video interface techniques.

**********

KIM - 1 / User Notes

I have run into a problem concerning use of the KIM interval timers. If this particular problem has not been addressed, here's what I have found:

Conclusion

An interval timer write operation does not work properly when that interval timer count is crossing zero at the time of the write.

Try the following simplified test on your KIM.

```
          LDA #NUM      A9 XX
          STA 1704      8D 0417
          LDA #F0       A9 F0
          STA 1707      8D 0717
(wait)    LDA 1707      AD 0717
          BPL (wait)    10 FB
          JMP KIM MON.  4C 4F1C
```

The divide by 1 interval timer address is loaded with a starting count "XX". Five machine cycles later, a long time period is loaded into the timer (F0 into 1707). The program waits for the long period to exhaust itself (~¼ sec) and then returns to the KIM monitor. Normally, the execution of this program will make the display blank for about ¼ second. However, if the number 05 is loaded in the first program steps (XX), the interval timer will not time out properly but will instead pass program flow immediately back to the KIM monitor. Now read the above conclusion again.

If your program using a KIM interval timer has appeared to fail occasionally, this may be the reason. The three KIMs I have tried all have this bug. Remember that the interval timers are always counting, and if one attempts a timer write at random times the write will be bad 1 out of 255 times on the average. Take the first two program lines out and verify that upon repeated manual random entries into the program the interval timer will occasionaly fail. (1:256 ave.)

One can get around this bug by simply doing two successive writes to the interval timer used. e.g.

```
LDA NUM
STA 1707
STA 1707
```

a) if the first STA was done at a bad time the next STA will be at a good time.

b) if the first STA was done at a good time the timer will also be OK at the second STA unless the first STA tries to load a 03 into a divide by one register. Therefore do not make the first STA involve 1704,170C,1744, or 174C. The second STA can then involve any timer register you want, to achieve the desired timing.

Timothy Martin
Argonne National Laboratory
Argonne, Illinois 60439

**********

HIGH SPEED CASSETTE INTERFACE

If Hypertape is beginning to seem slow, then you can now get one better. Ziptape will run at 4800 baud!

Of course you'll have to abandon the KIM cassette software and hardware to do it - that's the tradeoff.

Ziptape consists of a small p.c. board with one comparator chip on it and the associated load and dump software. It costs $26.50 and is available from Lew Edwards, 1451 Hamilton Ave.

It blows my mind to think that this little board with one I.C. on it can replace something like a Tarbell cassette interface for the S-100 folks.

Ziptape works fine at 4800 baud on my Sankyo ST-50 but Lew cautions that some recorders may only be able to handle 2400 or 3600 baud.

More info can be obtained by sending him an S.A.S.E.

**********

FORTH for the 6502 will be available in the not too distant future. An excellent article appeared in Doctor Dobbs Journal (May '78) which explained the principles of FORTH and gave several programming examples. This language seems ideal for micros because it's so compact and interfaces easily with assembly language. We'll be seeing more of FORTH for sure.

Want more info on FORTH?

An excellent manual is available for $5.00 from DECUS, 126 Parker St., Maynard, Ma 01754. Order FORTH Manual #11-232. This document contains enough implementation info to get a good idea of how it's constructed. If you only purchase one manual get the one from DECUS.

A Micro FORTH primer is available for $15.00 from Forth, Inc., 815 Manhattan Ave., Manhattan Beach, Ca 90266. This primer is a very good introduction to the language. Get the one for the 6800 as they don't have a 6502 version yet. These folks are into selling industrial versions of FORTH for several thousand dollars so don't expect any help for hobbyists with questions.

There is rumored to be a Forth newsletter from Forth Interest Group, 787 Old County Rd., San Carlos, Ca 94070.

**********

MEANWHILE.....

Are you wondering what's left from my equipment sale in the last issue? Everythings gone except the KIMSI, the two 8K memory boards, the 64x16 video board and the KIM enclosure.

That local user club in the San Fernando Valley area sure is active! Jim Zuber, club organizer, sent me the minutes of their last meeting.

If you're in that area and want to get in touch with this active group call Jim at 213-341-1610 or write him - 20224 Cohasset #16, Canoga Park, Ca 91306.

IN CLOSING...

Thats right, were moving again. (we are becoming moving experts) Brenda and I are really excited about the direction the newsletter is taking--we feel very positive that we'll be able to provide much better service to the 6502 fraternity. But we need YOUR support now more than ever. Let us know what direction you'd like to see our newsletter take.

MORE SOFTWARE? MORE HARDWARE? MORE ON HIGH LEVEL LANGUAGES?

MORE ON THEORY? MORE TEST REPORTS? MORE ON SYSTEM EXPANSION?

YOUR COMMENTS COUNT!